A Parallel/Distributed Platform for University Computational Infrastructure Service Provisioning[*]

Koulopoulos D.
Computer Systems Laboratory,
University of Patras
Patras, Greece
dkoulop@ee.upatras.gr

Goulas G.
Computer Systems Laboratory,
University of Patras
Patras, Greece
goulas@ee.upatras.gr

Papoutsis K.
Computer Systems Laboratory,
University of Patras
Patras, Greece
kpap@ee.upatras.gr

Housos E.
Computer Systems Laboratory,
University of Patras
Patras, Greece
housos@ee.upatras.gr

Abstract

The use of strongly interconnected computers, typically within a local area network environment, for the formation of computer clusters has been in use for several years with significant success and user acceptability. The availability of high bandwidth interconnection network among most University departments allows the creation of even larger such computer clusters. In this paper we present a system that can be used to effectively utilize the various computer resources that exist within a typical University environment for the formation of a parallel/distributed computing platform. The proposed system named PLEIADES allows its users to transparently utilize various computer resources in order to form virtual clusters for their parallel/distributed computational needs. In addition, PLEIADES can be also used as a computational infrastructure service provider (CISP) for automatic use by a remote computer process and in particular an application service providing (ASP) support scheme. The main design, architecture and implementation issues of the PLEIADES system are presented.

KEY WORDS: Parallel/Distributed Systems; Web Applications; XML; Application Service Provisioning; Internet Computing

1. Introduction

In a typical University environment the existence of a high-bandwidth interconnection network among its departments and research centers has given the opportunity for Internet-based parallel/distributed processing and remote application access and submission. These developments have been also fuelled by the fact that most computers are presently "always on and connected to the Internet" [1]. For the efficient utilization of the available computing resources, the distributed computing paradigm through the formation of "grid computers" [2] appears to be the winning strategy for producing effective and scalable solutions. Even though hardware is always getting faster, for several important classes of problems especially in the University community, such as simulation and scheduling, there is a need for larger machines and additional computing power in order to achieve better and faster solutions. Thus, the scalability offered by the distributed computing paradigm in conjunction with the utilization of idle machines around the globe makes this strategy extremely attractive.

The PLEIADES system described in this paper can be used as a computational infrastructure service provider whenever the required processing power is not available. In particular, whenever user applications require a specific set of machines, the PLEIADES system would provide a distributed environment for the selection, prioritization and

[*] Contract grant sponsor: Greek General Secretariat for Research and Technology, Program PENED 99.

Proceedings of the 4th International Workshop on Computer Science and Information Technologies CSIT'2002
Patras, Greece, 2002

assignment of resources to the user applications. In the current implementation, PLEIADES rely on the infrastructure of the Condor2 resource management system [3], [4]. Condor provides distributed processing services [5] and supports the message passing libraries PVM and MPI. PLEIADES as an open system has been designed in a manner that allows the use of other resource management systems that could appear in the future. The PLEIADES system has many similarities with projects in the areas of Meta-Computing and Internet Based Parallel Computing and all of the systems that allow access to remote resources through web based interfaces. Some of them are the Meta-Neos project [6], which is oriented towards optimization problems; the JAVADC [7], which provides a web-based framework for the execution and monitoring of distributed applications written in PVM, pPVM [8] and MPI; the WebFlow [9], which provides a general-purpose web-based visual interactive programming environment for distributed computing. WebSubmit [10] and UNICORE [11] aim on providing secure simplified and unified access to existing high performance computing systems.

For the remaining of the paper, there is a presentation of the PLEIADES system requirements, a detailed description of the PLEIADES prototype and all of its subsystems and a conclusions section.

2. System Functionality

The most essential requirement of PLEIADES involves the ability of users to form clusters of computers, which would then operate as networks of workstations (NOW) [12] and thus solve in a parallel/distributed manner computationally and/or memory intensive problems. In addition, PLEIADES which is designed to be an open computer system should be able to function as a Computational Infrastructure Service Provider (CISP) for the computational support of Application Service Providing (ASP) situations [13] which involves a computer to computer collaboration.

A necessary ingredient that the PLEIADES system must include is the ability of computer owners to donate CPU cycles to other users. The PLEIADES system should recognize a user type whose role is to allow the addition of his/her machine in the PLEIADES pool of machines and will be called donor-user for the remaining of this paper. The users that use the donated CPU cycles by executing their applications on the PLEIADES platform will be called acceptor-users or simply users. Another category of users required for various administrative and security tasks forms the administrator-users group of PLEIADES. Based on these requirements a high lever specification view of PLEIADES is shown in Fig. 1.

2.1 PLEIADES Services

PLEIADES provides the services shown in Figure 2 in order for the users to perform the functionalities described in the previous section and create the illusion of a virtual computer facility.

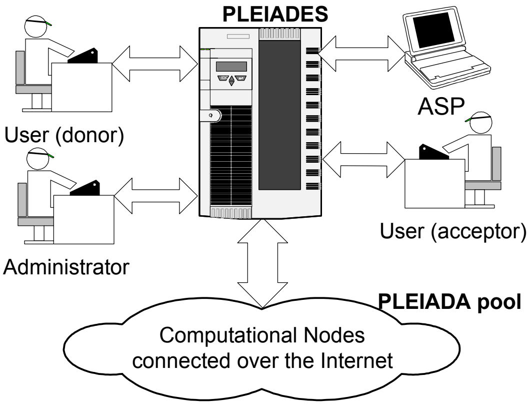

Fig. 1: The PLEIADES System

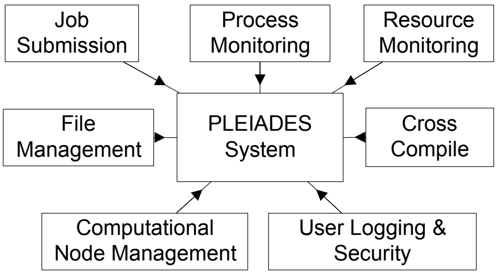

Fig. 2: PLEIADES services

➢ *Job submission* is the most critical service. It allows the users to submit their jobs for execution in distant machines by selecting the application executables and I/O files as well as the number and platform of the needed processors.

➢ *Resource and process monitoring services* maintain the status of each resource and process of the system. These services allow users to observe the status and progress of their work.

➢ *Computational Node Management service* supports the insertion of a workstation into a PLEIADES working pool, and provides the mechanism for its removal. It also allows a donor to define the conditions for the availability of his machine under the PLEIADES pool.

➢ *User Logging and Security services* are established for the various parts of the system. Each user is allowed to access only certain resources and services in accordance with the needs and permissions of the particular user group. The main

[2] The Condor Software Program (Condor) was developed by the Condor Team at the Computer Sciences Department of the University of Wisconsin – Madison. All rights, title, and interest in Condor are owned by the Condor Team.

security issue of such a system involves the full protection of the donor systems from malicious processes.

➢ *File management service* provides a separate file space for each user in which files can be transferred, edited and manipulated.

➢ *Cross-compile service* enables the user to compile the source files in distant machines.

2.2 iNOW Functionality

The use of the Internet as the interconnection medium for the formation of computer clusters [14], [6], [8] is more flexible and has a greater growth potential rather than LAN based Parallel / Distributed Processing, which was initially studied by the NOW Project [12], and later streamlined by the Beowulf project [15]. This type of Internet based cluster is named iNOW in this paper.

Applications must satisfy certain rules in order to attain significant benefits from an iNOW environment. If computers are interconnected by a high-bandwidth network that it is not overloaded when the virtual cluster is in actual use, the expected performance difference between the NOW and the iNOW case should be minimal. Thus, applications that use a LAN based NOW should perform satisfactorily in an iNOW environment if they have a significant amount of parallel computation load and are able to implement strategies for fault tolerance, dynamic cluster allocation and resizing.

PLEIADES using the services described in the previous section should allow for the creation of iNOW environments where users would be able to define the number and platform of required processors and memory capacity. The required binaries and input files for the application at hand should be transferred in the PLEIADES file space in advance and any required cross-compiling processing should be supervised by the user.

2.3 CISP/ASP Functionality

The use of the ASP approach for the solution of various problems is recently becoming mainstream due to the benefits and cost savings involved when various common and computationally intensive applications are maintained, shared, hosted and executed in a centralized environment [13]. In a University environment, that is the focus of our work, the existence of several university-wide Application Service Providing facilities in cooperation with PLEIADES would assist in avoiding the under-utilization of the computer resources and the better availability of various popular applications.

PLEIADES should have a simple and well-defined interface to all of its services that is designed for use by other computers acting without the assistance of actual human users. Through this interface, an ASP could submit its computational requests to PLEIADES, which then would act as a Computational Infrastructure Service Provider (CISP) for the various ASP situations that the University might create.

3. PLEIADES Architecture

The PLEIADES system prototype is designed using a tiered architecture, as shown in Fig. 3. Tier 1, the PLEIADES Front-End Tier (PFT), manages on the one hand the interaction with human users that submit their applications for execution and on the other hand the interface with computer users such as Application Service Providing organizations.

The second Tier, PLEIADES Middle Tier (PMT), provides an abstraction layer for the different resource managers that could be potentially used by the PLEIADES system. The third Tier, PLEIADES Resource Management Tier (RMT) provides the resource and job management services.

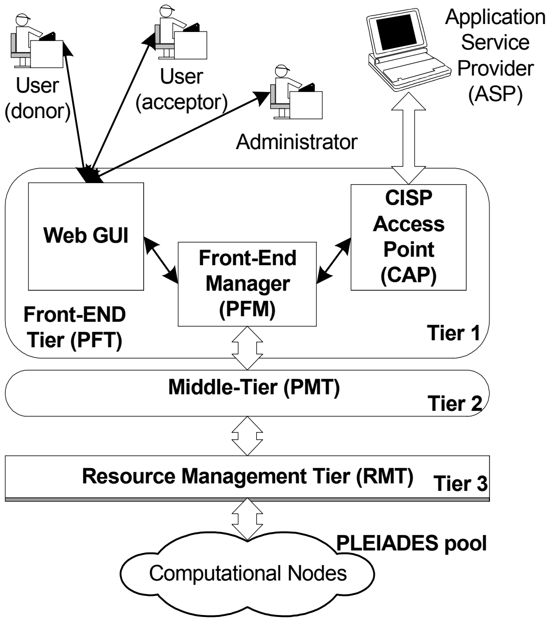

Fig. 3: The PLEIADES System Components

3.1 Front-End Tier (PFT)

The main component of the PFT is the PLEIADES Front-End Manager (PFM), which is assisted by the Web GUI and the CISP Access Point (CAP) interfacing subsystems. The Web GUI subsystem is intended for the human interactions and the CISP Access Point subsystem is used for the computer-to-computer interactions. The communication between the PFM and the interface subsystems utilizes XML based messages.

3.1.1 PLEIADES Front End Manager (PFM)

The PFM is a component designed to support multiple access points. PFM does not perform any authentication procedures and assumes that the access points perform this function. This design choice has been made because malicious behaviour must be blocked in very early stages, to avoid possible flaws and sideways that may exist in the data path.

The definition of the service requests and their validation is performed by the use of specific XML Document Type Definitions (DTD). In most cases, the PFM acts as a broker, forwarding the requests from the access points to the middle tier and back. In the file management case PFM actually performs the tasks. The directory structure that appears in the interface is actual directories. Fig. 4 presents the DTD of the file management service requests and Fig. 5 is an example of such a request. PFM uses the IBM XML4C API for the XML related issues.

```
<!ELEMENT filespace (userdata, cmd)>
<!ELEMENT userdata EMPTY>
 <!ATTLIST userdata username CDATA #REQUIRED>
 <!ATTLIST userdata password CDATA #REQUIRED>
<!ELEMENT cmd (mkdir | rmdir | mvdir | rmfile |
     dir | mvfile | touchfile | cat | store)>
<!ELEMENT mkdir EMPTY>
 <!ATTLIST mkdir dirname CDATA #REQUIRED>
<!ELEMENT rmdir EMPTY>
 <!ATTLIST rmdir DIRNAME CDATA #REQUIRED>
<!ELEMENT dir EMPTY>
 <!ATTLIST dir dirname CDATA #REQUIRED>
<!ELEMENT cat EMPTY>
 <!ATTLIST cat filename CDATA #REQUIRED>
<!ELEMENT store (#PCDATA) >
 <!ATTLIST store filename CDATA #REQUIRED>
<!ELEMENT rename EMPTY>
 <!ATTLIST rename oldname CDATA #REQUIRED>
 <!ATTLIST rename newname CDATA #REQUIRED>
<!ELEMENT rmfile EMPTY>
 <!ATTLIST rmfile filename CDATA #REQUIRED>
```

Fig. 4: DTD for file management service requests

```
<?xml version="1.0" ?>
<!DOCTYPE filespace SYSTEM "filespace.dtd">
<filespace>
    <userdata username="user" password="123" />
    <cmd>
        <mkdir dirname="test" />
    </cmd>
 </filespace>
```

Fig. 5: A request to create a directory named test

3.1.2 Web GUI

The Web GUI implements a web-based user interface for the PLEIADES system prototype. Through it, users can upload their executables and manage their file space. File operations include editing, copying, renaming, visualizing of input or output files and more. The Web GUI facilitates also the job submission service, with a selection of platforms for the available executables, and the other settings needed for a job submission. Upon submission, Job Monitoring can be achieved through the Web GUI subsystem. The Web GUI also provides interfaces to the cross-compile and to the donated machines management services. It is implemented in PHP, running under the Apache web server. Security is enforced using the widely adopted SSL standard. Authentication and session management functionalities are provided by the phpSecurePages component, which rely on the MySQL DBMS. This component also facilitates the user roles distinction since it supports different user levels. For the file management needs, the WebExplorer component is utilized, after some significant enhancements to meet PLEIADES needs.

3.1.3 CISP Access Point (CAP)

The Computational Infrastructure Service Provider (CISP) functionality is an innovative feature of PLEIADES and is provided through the CAP interface subsystem. Although an ASP external entity is modelled as a real user by the PLEIADES system, the interface style needed for such an interaction is extremely different than the Web GUI situation described in the previous section. The CAP interaction with the outside world is stateless and its clients submit their requests together with all the necessary credentials, and disconnects. Since an ASP may support multiple users, unknown to PLEIADES, thus the ASP is responsible for keeping all the state information of each user.

For the transfer of the requests, an XML-formatted message is transmitted over a POST / HTTP request. These requests, as stated before, include user credentials so that CAP can authenticate the users. The ASP administrators can pre-install their applications through the Web GUI, organize their directory structure as needed and then streamline the use of PLEIADES with the use of the CAP component. CAP is implemented as cgi-script, in C++, running under the Apache Web Server.

3.2 Middle-Tier (PMT)

The PLEIADES Middle Tier provides the services that are needed for the Tier 1 (PFT) to be able to actually perform their job and system management tasks. The most important services that the Middle Tier provides to PFT are job submission, job monitoring, job control, adding and deleting of hosts to the PLEIADES Pool.

The main reason for the existence of this Tier is the definition and implementation of a well-defined API for the communication of the PLEIADES Front-end Tier (PFT) with any particular implementation of the PLEIADES Resource Management Tier (RMT). This enables the PFM subsystem to be independent from the resource management system of Tier 3, since PMT hides all the interface and functional details of RMT.

The information regarding job submission, job status and job history is maintained in the PLEIADES Database. The communication between PMT and RMT has been fully customized for the Condor resource management system that is used in the present PLEIADES prototype.

3.3 Resource Management Tier (RMT)

For the current implementation of PLEIADES, the Resource Management Tier (RMT) is implemented using the Condor resource management system [3] [4]. One of the attractive characteristics of Condor is its ability to form efficient computer clusters from a given pool of available processors. The minimum requirement for a workstation to become a member of the PLEIADES pool is the installation of the client Condor software. This initial installation must be done by the system administrator for security, responsibility and liability related reasons. For PLEIADES when the Condor client process runs in a distant machine it runs with the permissions of the nobody account in order to ensure the integrity and security protection of the system. Condor does not require a Network File System (NFS) for data transfer. Condor cooperates with various message-passing protocols for the execution of parallel/distributed jobs.

Condor provides services for submitting, monitoring and controlling serial or parallel jobs. In order for a job to specify its execution environment requirements, as they relate to the platform needed and number of processors desired, a special submit file is created at Tier 2. This file contains all the parameters required by Condor to correctly manage and schedule the particular job. Based on these requirements Condor performs a sophisticated matching between the job requirements and the available resources and tries to perform the best assignment [16].

Condor requires one machine to be the master of the PLEIADES computer pool. This machine called Condor Master, contains the full Condor System and keeps track of all the running jobs, the available machines and their status. For each Computational Node (CN), Condor provides all the necessary functionality in order for the owner of the machine to define when the particular machine will be available for PLEIADES. For all parallel/distributed jobs the master process must run on designated machines and if the executables for the particular machines are not available the user can create them by utilizing the PLEIADES cross-compiling service.

Conclusions

A system, named PLEIADES, for the creation of virtual networks of workstations that exploits the existence of a high-bandwidth interconnection network among most universities of Greece has been presented. PLEIADES utilizes the simplicity and uniformity that Internet has created in order to make the sharing of university-wide resources feasible and productive. The performance of virtual clusters that was observed was comparable to LAN based Network of Workstations if the network was slightly loaded and the applications were amenable to parallel/distributed computation.

In addition, the possibility of using the PLEIADES system for the execution of typical serial applications in need of significant computing resources and are submitted either from an Application Service Providing organization (ASP) or an actual human user has been explored. The use of XML for the exchange of the appropriate information between the ASP and PLEIADES and among the various PLEIADES internal subsystems, gives additional flexibility and module independence to the design. The PLEIADES Middle Tier (PMT) allows the system to easily incorporate new resource management systems as they become available, without requiring change of the first tier.

The special nature of the donated computer use on distant machines, that is mainly characterized by the fact that the computer owner can reclaim the use of the machine at any time, requires the use of a message passing library that allows for such a removal to occur without significant loss of work. Moreover, it should be the case that if a single computer is reclaimed by its owner, the remaining cooperating processes running on the remaining computers of the specific virtual cluster will not also terminate abruptly. For the present resource manager used in PLEIADES, this fact points to the use of the PVM message-passing library for the creation of the parallel/distributed applications, although MPI based applications give better performance results when special multiprocessor machines are also involved in the cluster.

Additional use of PLEIADES for research and development purposes is planned. The new training and the special tools needed for a parallel/distributed platform to be effectively used, is in many cases the basic obstacle for the extensive use of this type of systems. However, the use of PLEIADES as a resource management system for the scientific community even when traditional applications are involved appears possible and extremely desirable.

References

[1] "Always on: Living in a networked world", *IEEE Spectrum* 2001;38 (1)

[2] I. Foster and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure". Morgan-Kaufmann, 1999

[3] J. Basney and M. Livny, "Deploying a High Throughput Computing Cluster", in *High Performance Cluster Computing*, Rajkumar Buyya (ed), Vol. 1, Chapter 5, Prentice Hall PTR, May 1999

[4] M. Livny, J. Basney, R. Raman, and T. Tannenbaum, "Mechanisms for High Throughput Computing", *SPEEDUP Journal* 1997; 11(1)

[5] J. Pruyne and M. Livny. "Providing Resource Management Services to Parallel Applications", *Proceedings of the Second Workshop on Environments and Tools for Parallel Scientific Computing*, May, 1994

[6] J.P. Goux, S. Kulkani, J. Linderoth and M. Yoder. "An Enabling Framework for Master-Worker Applications on the Computational Grid". *HPDC 2000 Conference.* Available from http://www.mcs.anl.gov/metaneos/papers/mw2.ps

[7] K. Maly, P.K. Vangala and M. Zubair. "JAVADC: A Web – Java Based Environment to Run and Monitor Parallel Distributed Applications". Technical Report. Old Dominion University, 1997

[8] K. Maly, M. Zubair, Shubhangi U. Kelkar. "*A Parallel and Distributed Computing Environment for Scientific Applications*". PhD Thesis, 1996

[9] D. Bhatia, V. Burzevski, M. Camuseva, W. Furmanski and G. Premchandran, "WebFlow – a visual programming paradigm for Web/Java based coarse grain distributed computing". *Workshop on Java for Computational Science and Engineering*, Syracuse University, December, 1996

[10] R. McCormack, J. Koontz and J. Devaney. "Seamless Computing with WebSubmit", *Concurrency: Practice, and Experience* 1999; 11(15)

[11] J. Almond, D. Snelling, "UNICORE: Secure and Uniform Access to Distributed Resources via the World Wide Web". White Paper, October, 1998

[12] Anderson, T., Culler, D., and Patterson, D., "A Case for NOW (Networks of Workstations)", *IEEE Micro*, 1995

[13] "ASP – Application Service Providing, The Ultimate Guide to Hiring rather than Buying Applications". Vieweg, 2000

[14] M. Good and J.P. Goux, "iMW : A Web-based Problem Solving Environment for Grid Computing Applications". MetaNEOS Project Technical Report, available at http://www.mcs.anl.gov/metaneos/papers/imw.ps

[15] The Beowulf Project site, at http://www.beowulf.org

[16] R. Raman, M. Livny, and M. Solomon, "Resource Management through Multilateral Matchmaking", *Proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9)*, Pittsburgh, Pennsylvania, August, 2000, p 290-291